Citizen Management Application

Object-Oriented Programming - Project Report

Nguyen Mai Anh – 20215178

Cao Huy Dong – 20210188

Nguyen Tan Dung – 20215186

2024-06-06

1. Introduction

1.1. Background

The escalating challenges of population growth, coupled with an increasing number of temporary residents and vacant homes, underscore the pressing need for an effective residential management system. This project addresses the complexities of managing a burgeoning population within densely populated residential areas. As the population continues to rise annually, the surge in temporary residents and vacant residences poses unique challenges for community administrators.

1.2. Objective

The primary objective of this project is to develop a a specialized application for managing neighborhood information and fees, specifically designed for the convenience of administrators. The application aims to allow administrators to easily access and update resident information while efficiently managing the fee collection process for individual households.

1.3. Scope

Key features of the application include the ability to view and edit residents' personal information, including community event details and important notifications. Additionally, administrators can add, modify, and track fees for each household, facilitating financial management within the community.

The application's interface is designed to be simple, user-friendly, and focused on management functions tailored for administrators. The project's objective is to provide an efficient and lightweight solution, enabling administrators to carry out community information management and fee collection tasks with utmost effectiveness.

1.4. Technology Used

To build a robust and efficient system, aside from Java, the following technologies will be employed:.

-MVC Architecture

-JavaFX for the Frontend

-PostgreSQL for Database management.

2. Overall Description

2.1 Purpose and Problem Requirements

***Information regarding household and population is recorded and managed by the* neighborhood head as follows:**

• Details about individuals in each household are stored in a family register with a unique identification number. The family register includes information such as household number, head of household's name, house number, street (hamlet), ward (commune, town), and district (district, township).

• The family register provides specific details for each individual in the household, including name, alias, date of birth, place of birth, hometown, ethnicity, occupation, workplace, ID card or Citizen Identification Card (CCCD) number, date and place of issue, date of permanent residence registration, and previous permanent residence address if applicable. If the individual is the head of the household, their information will be recorded on the first page; otherwise, additional details such as the relationship to the head of the household will be added.

• Changes in population details include:

• Adding new individuals: When a family has a new member, details such as occupation, ID card, and new permanent residence information will be left blank, marked as "newborn."

• Updating individual information: If an individual move elsewhere, details like date of departure, destination, and notes will be added. In case of death, the note will state "Deceased."

• Changes related to the entire household (e.g., changing the head of the household) need to be recorded, specifying the nature of the change and the date.

• When a household is separated from an existing family register, a new family register will be created for the selected individuals.

• If a family member is away for an extended period, they must notify the neighborhood head and request a temporary absence certificate. Conversely, if someone from another locality resides temporarily for a certain period, they must declare to be issued a temporary residence certificate.

• Additionally, the neighborhood head and deputy have the need to quickly search for information, view the change history of a household, and periodically compile statistics based on criteria such as gender, age group, time frame, and record temporary absence/temporary residence.

Fee Collection Management:

- Annually, the neighborhood committee collects fees and contributions from households, managed by the accountant. Sanitation fees are mandatory for all households, collected once a year at the rate of 6,000 VND per person per month.

- The accountant compiles a list of households and corresponding population numbers. Subsequently, they visit each household to collect fees and record the amount paid. Contributions, on the other hand, vary for each household and depend on different fundraising events, such as supporting veterans/martyrs on July 27th, supporting Children's Day, contributing to the poor, assisting flood-affected compatriots, etc.

- The accountant also needs to summarize the total amount collected in each event, the total number of households that have paid, and view the details of each household's contributions.

2.2 Process Analysis

We deal with this problem though some processes

2.2.1 Problem-solving processes

Population and Household Management

Basic information for the problem-solving processes includes:

- Personal information
    - Full name, citizen identification number, date of birth, gender, ethnicity, religion, hometown, etc
    - Educational background, occupation
    - Residence, phone number, email, etc
- Household information
    - Head of household's name, household ID, members, address, etc.
    - Family relationship: Father, mother, spouse, children.
    - Information about household members: Number of people in the household, details about each member.
- Temporary residence report information
    - Temporary residence report ID, ID of person moving
    - Place, time and reason why person need to move
- Temporary absence report information
    - Temporary absence report ID, ID of person absent
    - Place, time, mobile phone and reason why person need to move
- Death report information
    - Death report ID, ID of death person, ID of person reporting
    - Time for death, time creating report and reason why people die.

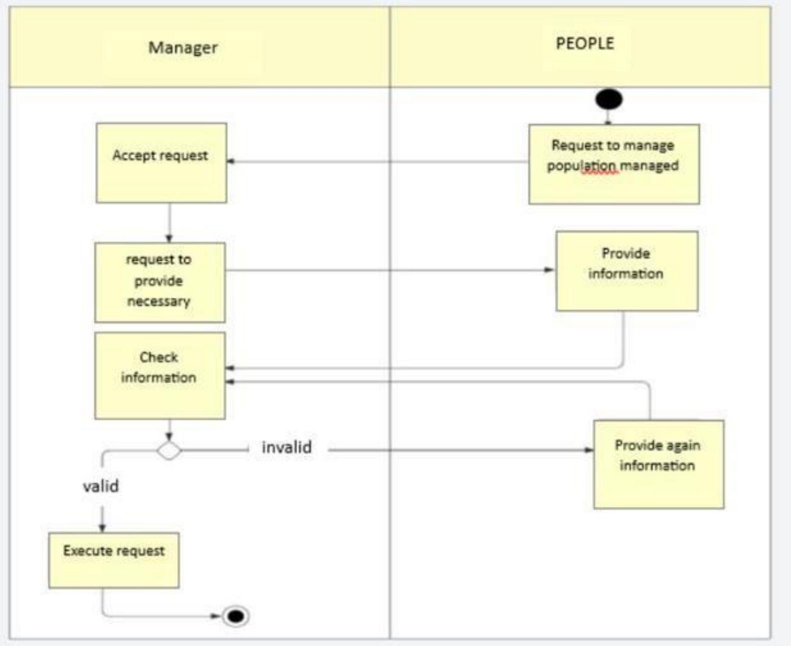
Fee collection Management

- Contributor information
    - Full name: Contributor's full name.
    - ID card/Citizen Identification Card (CCCD) number: Contributor's ID card or citizen identification number.
    - Contact information: Contributor's address, phone number, email.
- Fee type information
    - Fee classification: Distinguish between mandatory fees and contributions.
    - Contribution purpose: For contributions, specify the specific goal that the contribution is directed towards.
    - Information about the fee collection.
- Fee amount
    - Specific amount.
    - Date of fee collection: The time when the fee is collected.
- Contribution and Payment History
    - Contribution history: History of previous contributions and payments by the contributor.

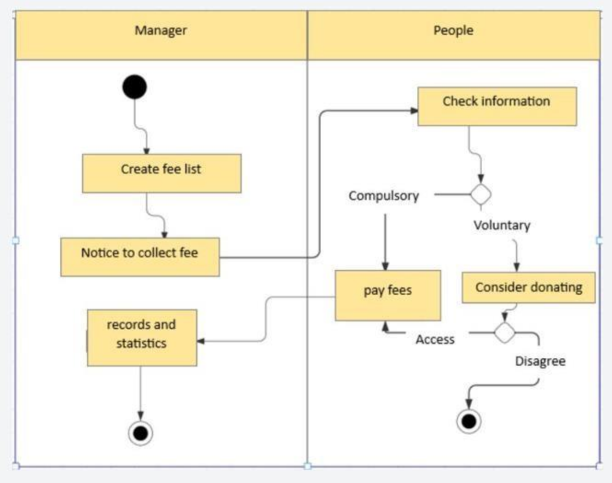
This information aids in managing and tracking fees transparently and efficiently, from identifying the type of fee to managing them.

2.2.2 Process Description Diagram

Activity Diagram
Population and Household Management



Fee Collection Management



2.3 Functional Requirements

- Login functionality:
  - Purpose: Allows administrators to access the system. Security measures must be in place to prevent unauthorized access, requiring users to provide a username and password.
  - User Authentication: Usernames and passwords must be securely stored in the system. The interface should be user-friendly and provide clear error messages for incorrect login attempts.User Requirements:
    - Usernames must be unique and at least 6 characters long.
    - Passwords must be at least 8 characters long, including uppercase letters, special characters, lowercase letters, and numbers.
- Population management functionality:
  - Registration: Handles permanent and temporary residence registration, modifications, and deletions of individual information.

    - Statistics and Reporting: Provides statistical data and reports on the population status.
    - Data Accuracy: Ensures the accuracy, completeness, and timeliness of population information.
    - Stored Information: Includes storing data such as full name, date of birth, ID card number, gender, religion, occupation, and address.
- Household Management Functionality
    - Household Actions: Enables actions like adding new households, splitting households, modifying household information, and deleting households.
    - Statistics: Provides statistics on the number of households, permanent addresses, and members.
- Fee and Contribution Management Functionality
    - Create/Edit/Delete Categories: Allows the creation, modification, and deletion of fee and contribution categories.
    - Create/Edit/Delete Payers: Manages the entities responsible for payments.
    - Information Storage: Stores information about fees and contributions, including payer names and addresses, securely and safely.

3. Use Case

3.1. Actors

This section discusses the actors of the system

3.1.1. User

Users are the primary actors who interact with the system. They can log in, log out, and sign up. Once registered, they can view and search for properties. Users include both population managers and fee collection managers who have specific roles and responsibilities within the system.

3.1.2. Population manager

Population managers inherit the general capabilities of a User, such as logging in, logging out, and signing up. In addition to these capabilities, Population Managers are responsible for managing citizens and their properties. They oversee the system operations, ensuring that all aspects of the property management are running smoothly. Their responsibilities include:

- Compile populational statistics
- Manage household registration
- Manage population registration

3.1.3.Fee Collection manager

Fee Collection Managers also inherit the basic functionalities of a User, including the abilities to log in, log out, and sign up. Their role is focused on the financial aspects of the system. They are responsible for collecting fees associated with property transactions. Their tasks include:

- Managing venue items of population in your area.

3.2. Use Case Diagram

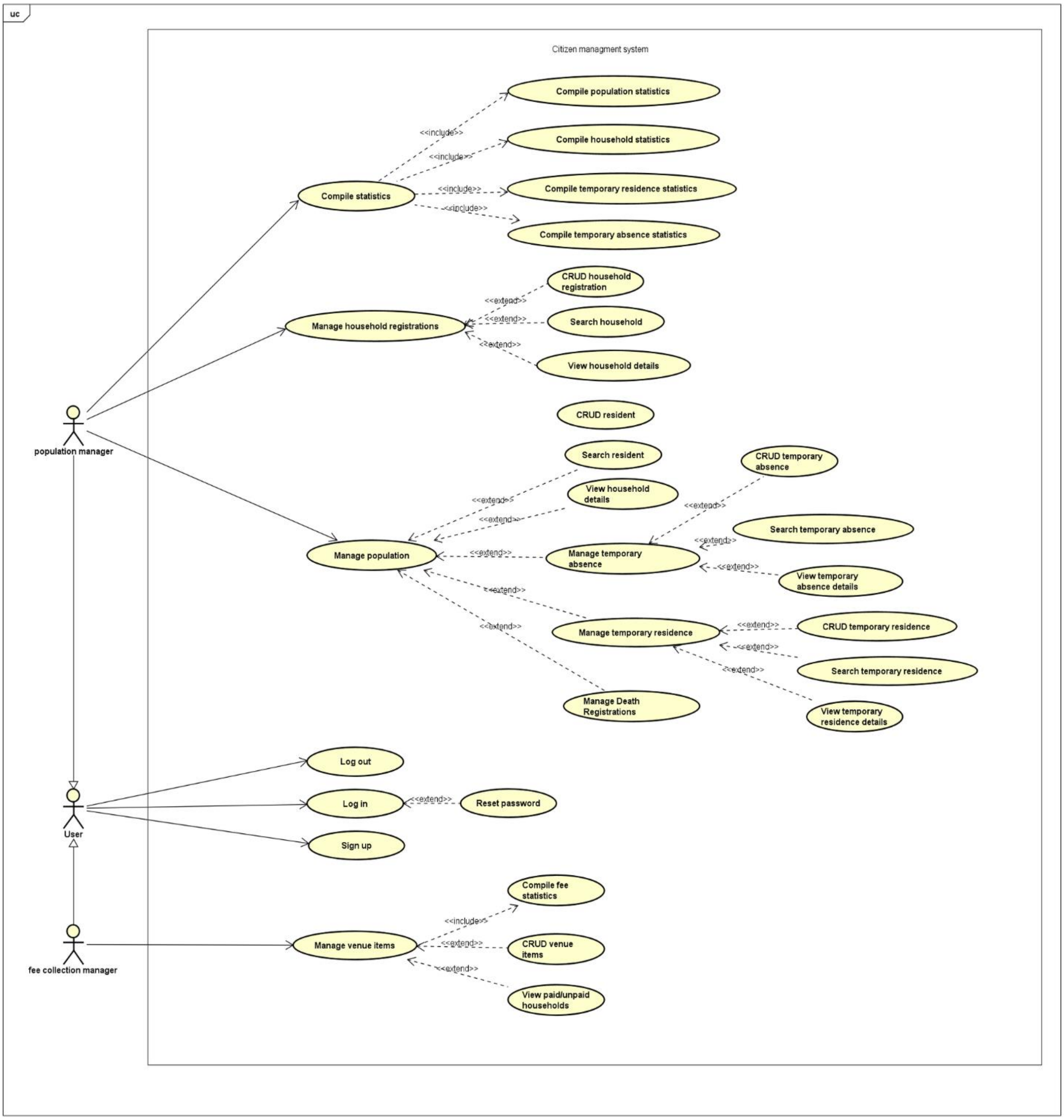


Figure 1: Use Case Diagram

4. Design

4.1. Architecture

The application is built using MVC Framework/ Architecture, an architectural pattern that separates an application into three main logical components: the model, the view, and the controller. Each of these components are built to handle specific development aspects of an application.

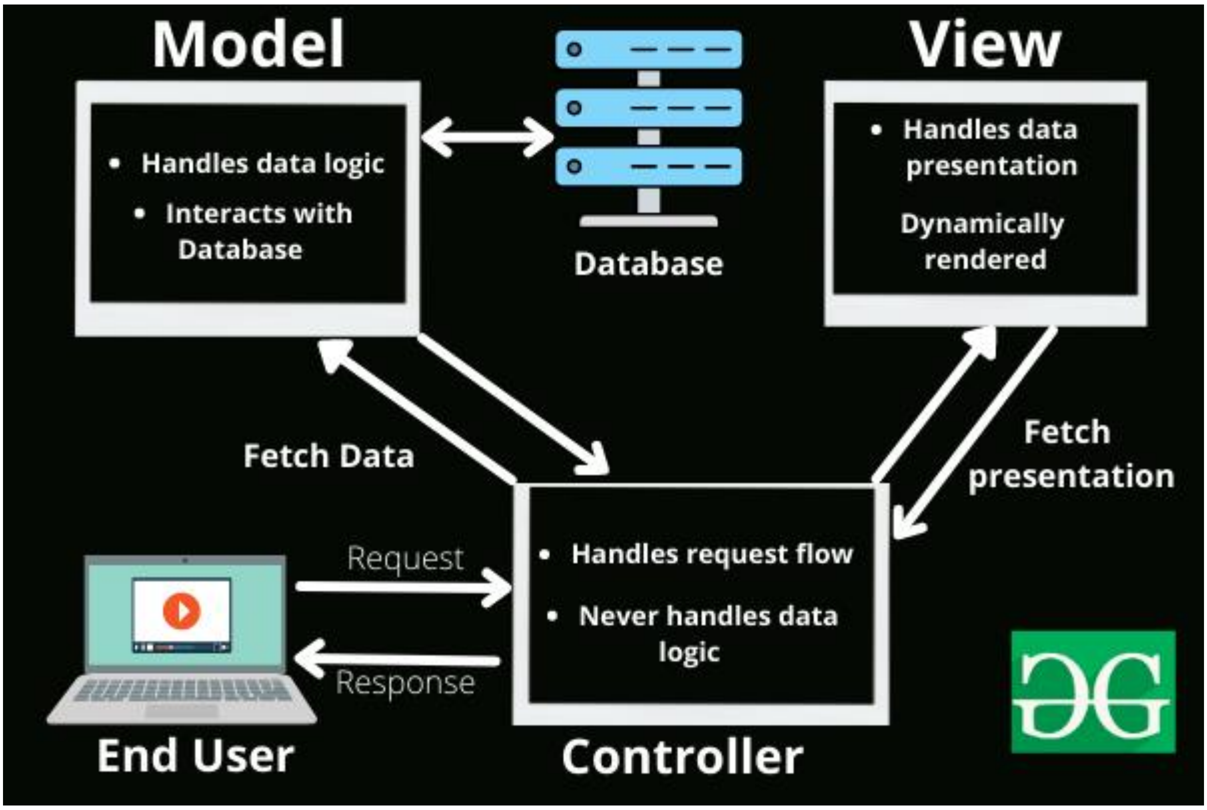


Figure 2: MVC Architecture

Model

-The Model component corresponds to all the data-related logic that the user works with. This can represent either the data that is being transferred between the View and Controller components or any other business logic-related data. For example, a Customer object will retrieve the customer information from the database, manipulate it and update it data back to the database or use it to render data.

View

-The View component is used for all the UI logic of the application. For example, the Customer view will include all the UI components such as text boxes, dropdowns, etc. that the final user interacts with.

Controller

-Controllers act as an interface between Model and View components to process all the business logic and incoming requests, manipulate data using the Model component and interact with the Views to render the final output. For example, the Customer controller will handle all the interactions and inputs from the Customer View and update the database using the Customer Model. The same controller will be used to view the Customer data.

4.2. Database Design

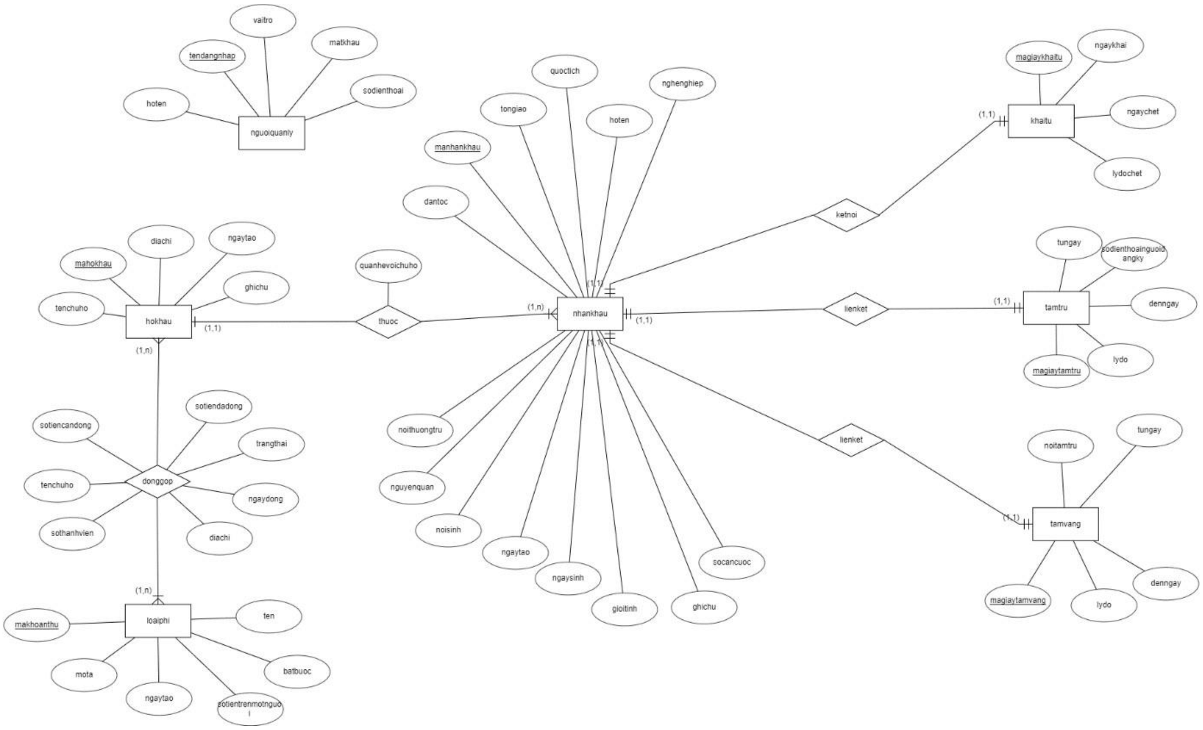
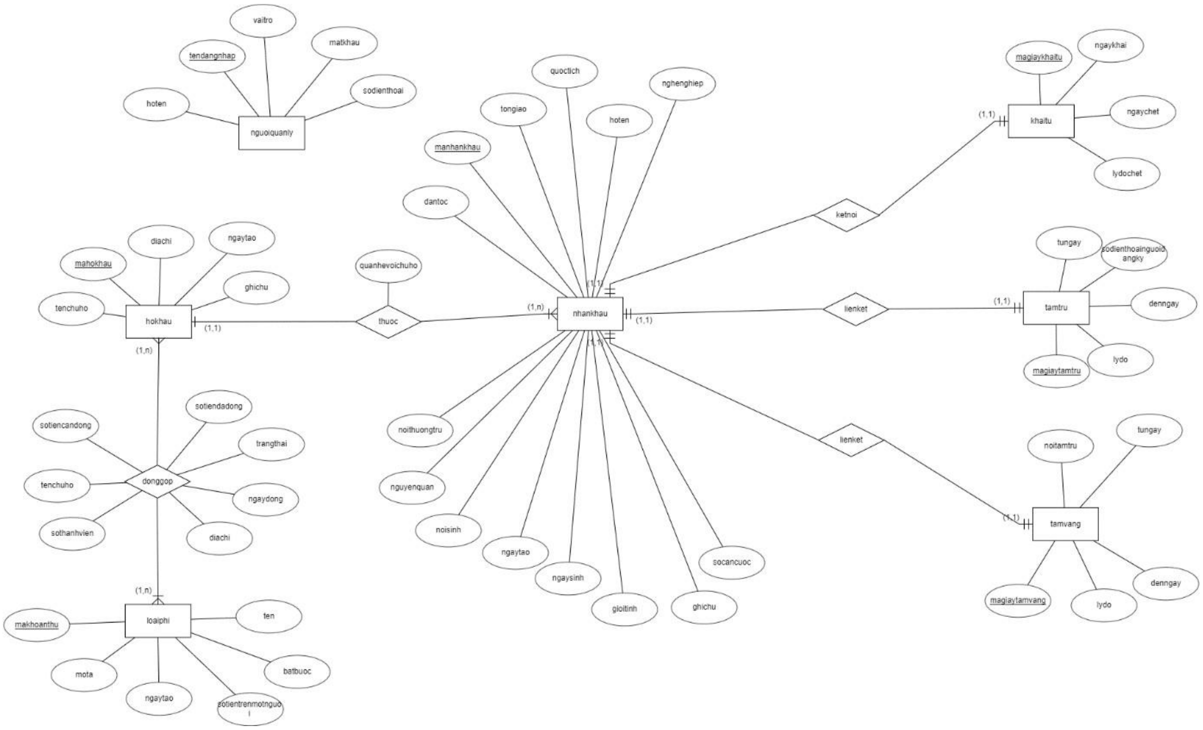
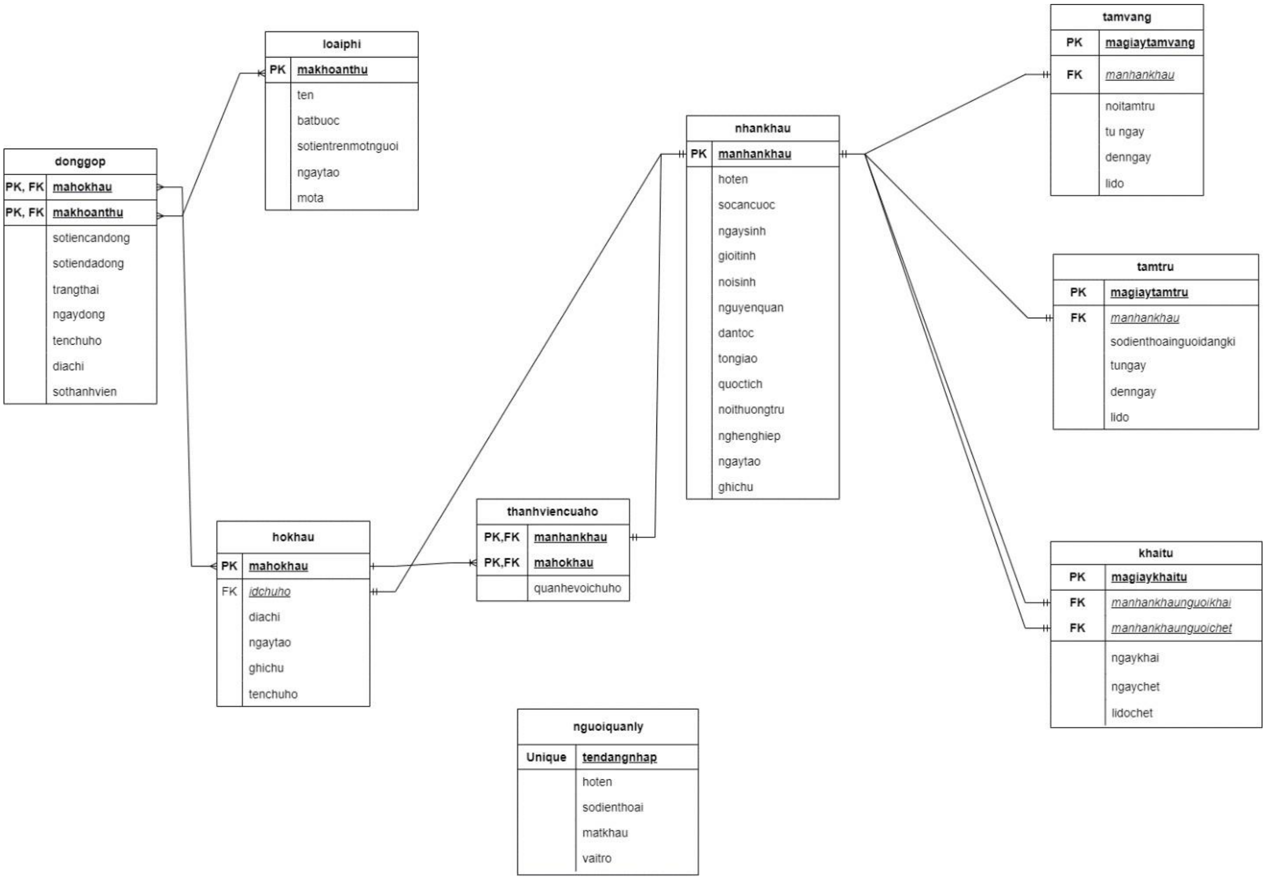


Figure 3: Entity-Relationship Diagram

4.2.1 Entity Relationship diagram





4.2.2 Data descriptions of table:

4.2.2a Table NGUOIQUANLY

Attribute	Data Type	Constraints	Note
HOTEN	VARCHAR(50)	NOT NULL	
TENDANGNHAP	VARCHAR(50)	UNIQUE, NOT NULL	
MATKHAU	VARCHAR(20)	NOT NULL	
SODIENTHOAI	CHAR(15)	NOT NULL	
VAITRO	BOOLEAN	NOT NULL	True – citizen manage; False – fee manage

4.2.2.b Table NHANKHAU

Attribute	Data Type	Constraints	Note
MANHANKHAU	INT	PRIMARY KEY, SERIAL	
HOTEN	VARCHAR(50)	NOT NULL	
SOCANCUOC	VARCHAR(15)		
NGAYSINH	DATE	NOT NULL	
GIOITINH	BOOLEAN	NOT NULL	True – Male, False – Female
NOISINH	VARCHAR(200)	NOT NULL	

NGUYENQUAN	VARCHAR(200)	NOT NULL	
DANTOC	VARCHAR(20)	NOT NULL	
TONGIAO	VARCHAR(20)	NOT NULL	
QUOCTICH	VARCHAR(20)	NOT NULL	
NOITHUONGTRU	VARCHAR(200)		
NGHENGHIEP	VARCHAR(100)		
NGAYTAO	DATE		
GHICHU	VARCHAR(20)		

4.2.2c Table TAMTRU

Attribute	Data Type	Constraints	Note
MAGIAYTAMTRU	INT	PRIMARY KEY, SERIAL	
MANHANKHAU	INT	PRIMARY KEY	
SODIENTHOAINGUOIDANGKY	VARCHAR(15)	NOT NULL	
TUNGAY	DATE	NOT NULL	
DENNGAY	DATE	NOT NULL	
LYDO	VARCHAR(300)		

4.2.2d Table TAMVANG

Attribute	Data Type	Constraints	Note
MAGIAYTAMVANG	INT	PRIMARY KEY, SERIAL	
MANHANKHAU	INT	PRIMARY KEY	
NOITAMTRU	VARCHAR(300)	NOT NULL	
TUNGAY	DATE	NOT NULL	
DENNGAY	DATE	NOT NULL	
LYDO	VARCHAR(300)		

4.2.2e Table KHAITU

Attribute	Data Type	Constraints	Note
MAGIAYKHAITU	INT	PRIMARY KEY, SERIAL	
MANHANKHAUNGUOIKHAI	INT	PRIMARY KEY	
MANHANKHAUNGUOICHET	INT	PRIMARY KEY	
NGAYKHAI	DATE	NOT NULL	
NGAYCHET	DATE	NOT NULL	
LYDOCHET	VARCHAR(300)		

4.2.2f Table HOKHAU

Attribute	Data Type	Constraints	Note
MAHOKHAU	INT	PRIMARY KEY, SERIAL	
ID CHUHO	INT	NOT NULL	
DIACHI	VARCHAR(200)	NOT NULL	
NGAYTAO	DATE		

GHICHU	NVARCHAR(200)		
TENCHUHO	NVARCHAR(300)		

4.2.2g Table THANHVIENCUAHO

Attribute	Data Type	Constraints	Note
MANHANKHAU	INT	PRIMARY KEY	
MAHOKHAU	INT	PRIMARY KEY	
QUANHEVOICHUHO	VARCHAR(100)	NOT NULL	

4.2.2h Table LOAIPHI

Attribute	Data Type	Constraints	Note
MAKHOANTHU	INT	PRIMARY KEY, SERIAL	
TEN	VARCHAR(50)	NOT NULL	
BATBUOC	BOOLEAN	NOT NULL	False – not required True – required
SOTIENTRENMOTNGUOI	BIGINT	NOT NULL	
NGAYTAO	DATE		
MOTA	NVARCHAR(300)		

4.2.2i Table DONGGOP

Attribute	Data Type	Constraints	Note
MAHOKHAU	INT	PRIMARY KEY	
MAKHOANTHU	INT	PRIMARY KEY	
SOTIENCANDONG	BIGINT		
SOTIENDADONG	BIGINT		
TRANGTHAI	BOOLEAN	NOT NULL	True – submitted False – not submitted
NGAYDONG	DATE		
TENCHUHO	NVARCHAR(300)		
DIACHI	NVARCHAR(300)		
SOTHANHVIEN	INT		

4.2.3 Relationship between tables:

- Table NHANKHAU – HOKHAU: Through table THANHVIENCUAHO (relationship type 1-many: 1 household can conclude some people)
    - Foreign Key: THANHVIENCUAHO.MANHANKHAU (references NHANKHAU.MANHANKHAU)
    - Foreign Key: THANHVIENCUAHO.MAHOKHAU (references HOKHAU.MAHOKHAU)
    - This establishes relationships between individuals in NHANKHAU and households in HOKHAU, defining family relationships.

- Table NHANKHAU – TAMTRU: the foreign key relationship (relationship type 1-1: 1 person can move to 1 temporary place)
    - Foreign Key: TAMVANG.MANHANKHAU

    (references NHANKHAU.MANHANKHAU)
  - This indicates that the TAMVANG table stores information about temporary absence from the registered residence for individuals in the NHANKHAU table.

- Table NHANKHAU – TAMVANG: the foreign key relationship (relationship type 1-1: 1 person can go far away 1 hometown)
  - Foreign Key: TAMVANG.MANHANKHAU (references NHANKHAU.MANHANKHAU)
  - This indicates that the TAMVANG table stores information about temporary absence from the registered residence for individuals in the NHANKHAU table.
- Table NHANKHAU – KHAITU: two foreign key relationships (relationship type 1-1: 1 person died 1 time)
  - Foreign Key: KHAITU.MANHANKHAUNGUOIKHAI (references NHANKHAU.MANHANKHAU)
  - Foreign Key: KHAITU.MANHANKHAUNGUOICHET (references NHANKHAU.MANHANKHAU)
  - This represents declarations related to death of individuals in the NHANKHAU table.
- Table LOAIPHI – HOKHAU: through table DONGGOP (relationship many – many: many people can contribute 1 fee and 1 people can contribute many fees)
  - Foreign Key: DONGGOP.MAKHOANTHU (references LOAIPHI.MAKHOANTHU)
  - FOREIGN KEY (MAKHOANTHU)
    REFERENCES LOAIPHI(MAKHOANTHU)

5. Class Design

5.1. General Class Diagram

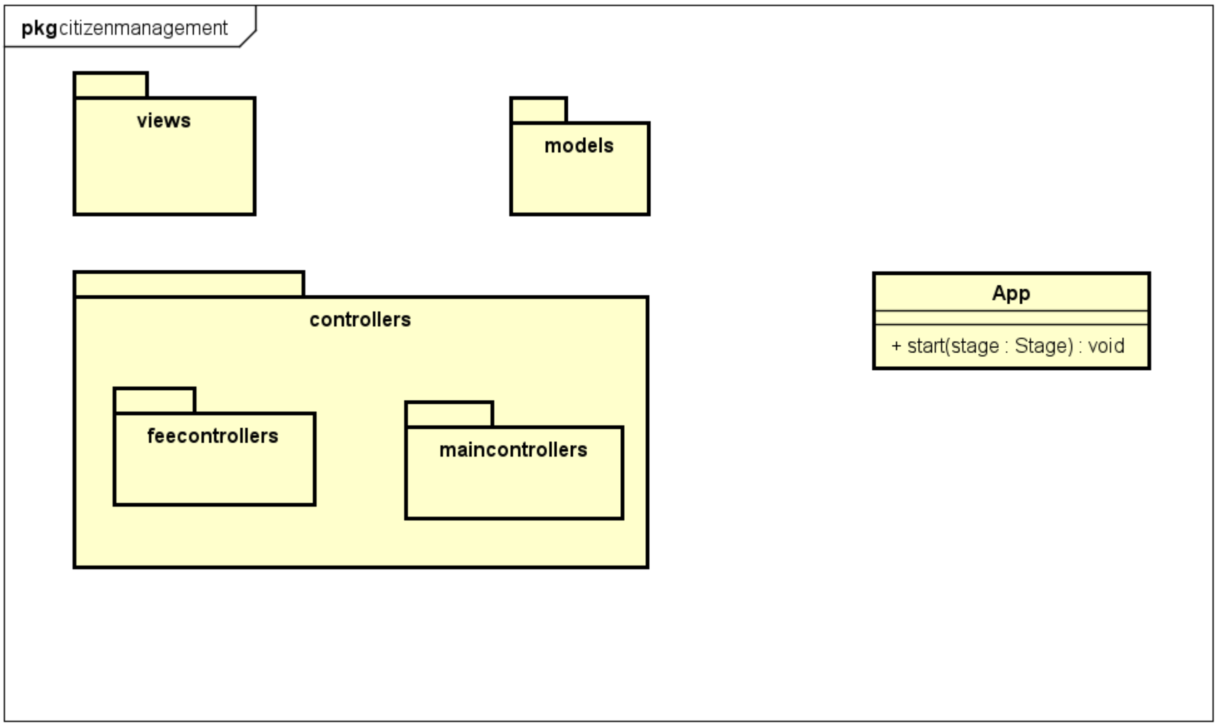


Figure 4: General Class Diagram

5.2. Model Class Diagram

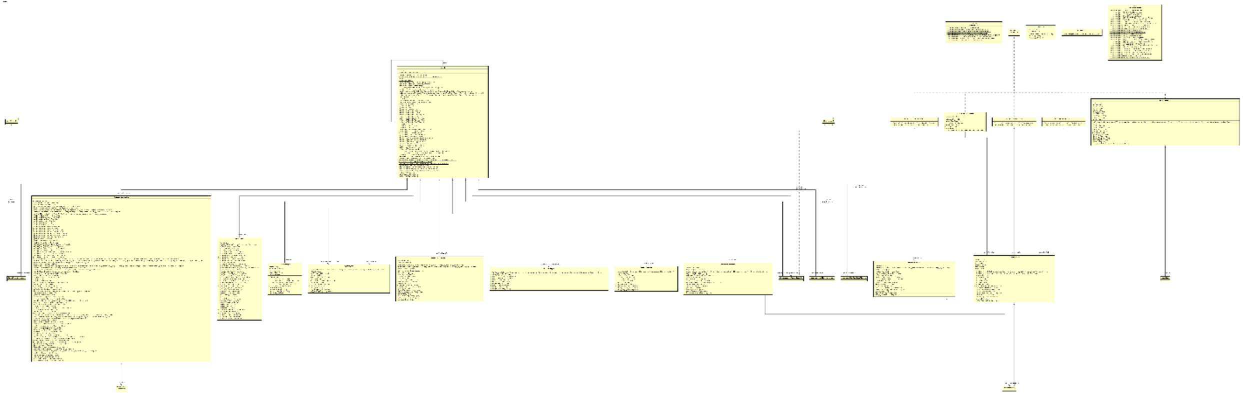


Figure 5: Model Class Diagram

5.3. View Class Diagram

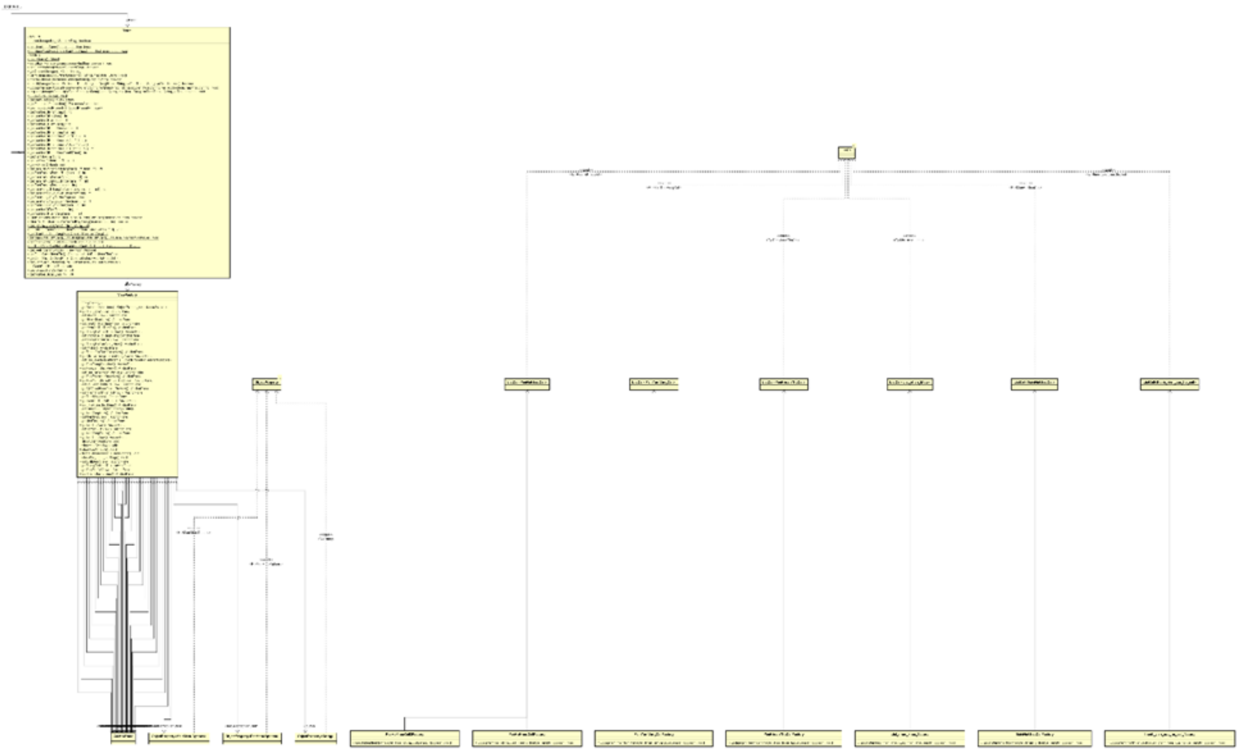


Figure 6: View Class Diagram

5.4. Controller Class Diagram

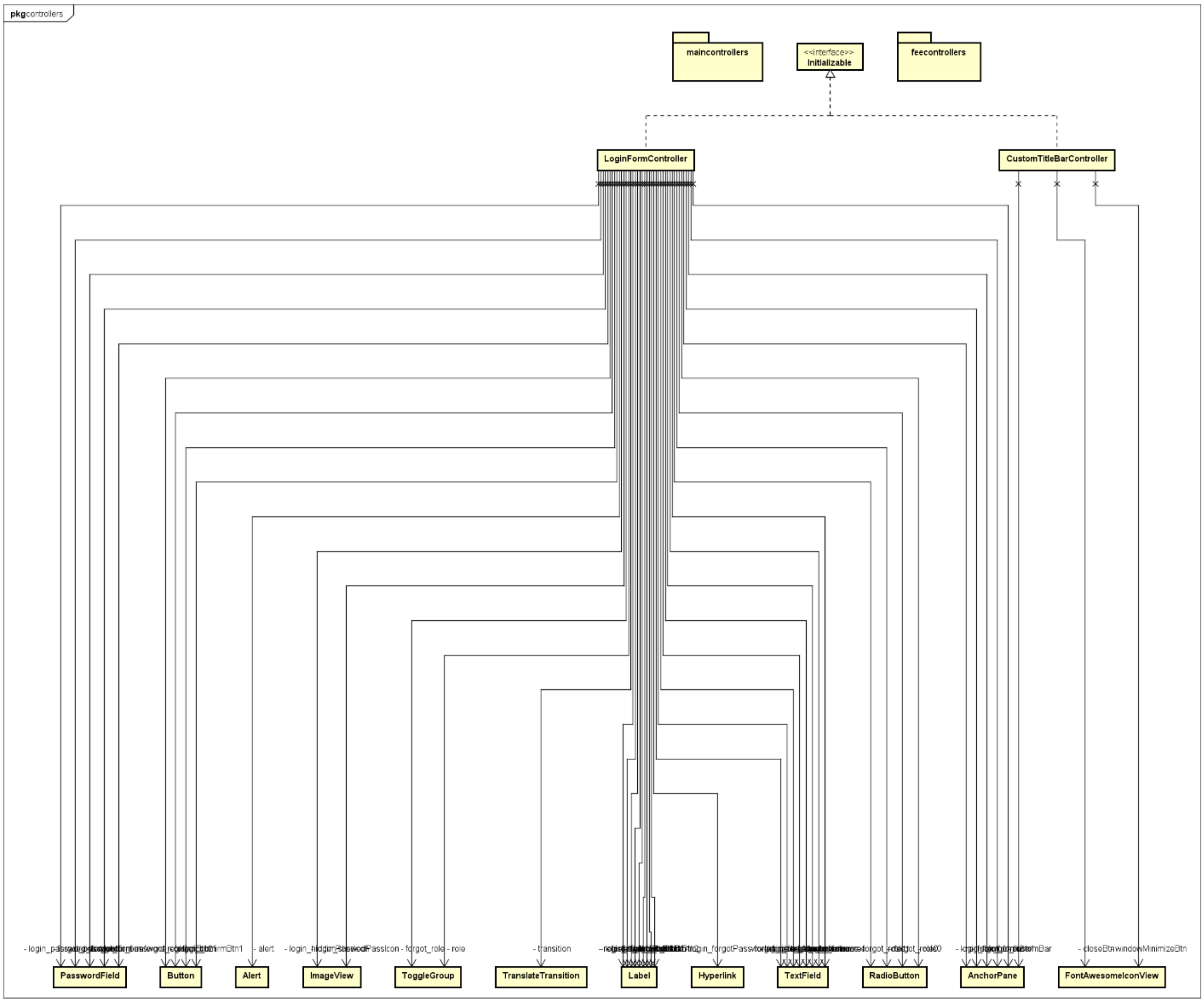


Figure 7: Controller Class Diagram

5.4.1 Fee Controllers

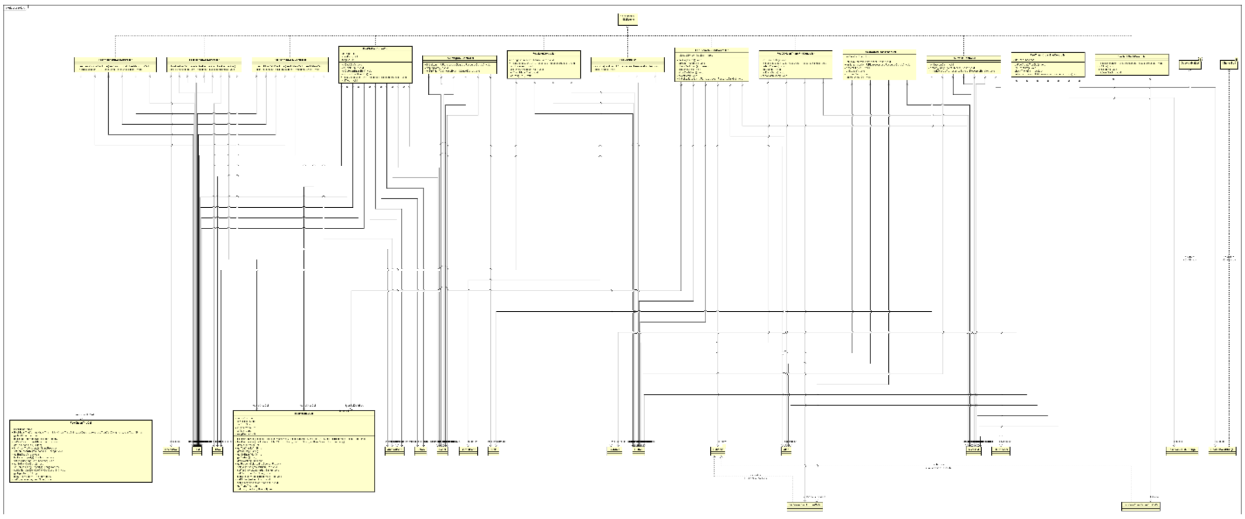


Figure 8: Fee Controller Class Diagram

5.4.2 Main Controllers

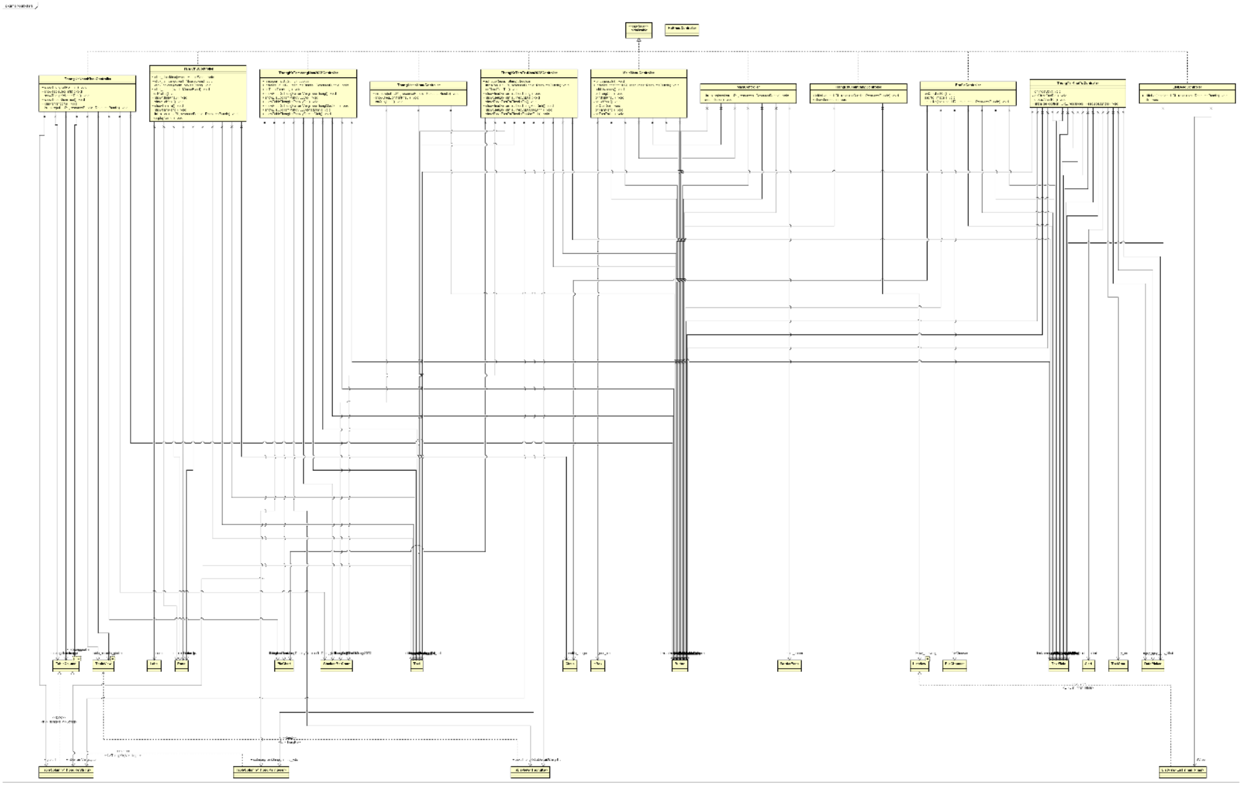


Figure 9: Main Controller Class Diagram

6. Team Member Assignments

- Nguyen Mai Anh - Backend
- Cao Huy Dong - Frontend
- Nguyen Tan Dung - Database